



DEVOPS E INTEGRAÇÃO CONTÍNUA

AULA 1



Prof. Mauricio Antonio Ferste



CONVERSA INICIAL

Durante todo o processo de crescimento mundial, passamos por vários momentos que promoveram grandes mudanças no contexto de desenvolvimento. Desde a década de 1940, quando o computador se tornou parte fundamental na estratégia de guerra; passando pelos anos seguintes, quando ele afirmou-se importante para promover diferenças competitivas para as empresas, vários processos foram surgindo e nos direcionaram para aprimoramentos e otimização da tecnologia da informação (TI).

Temos, com isso, que lidar com questões como:

- **Como melhor tratar a TI nas empresas?**
- **Como a área de TI pode nos tornar mais fortes, competitivamente falando?**

Logicamente, esse processo não é simples. É consequência de diversos processos e pressões que ocorreram com o tempo e que sempre culminavam com o contexto de melhor e maior competitividade, na área de TI. **E eis que surge o DevOps!**

O DevOps, uma fusão das palavras *desenvolvimento* e *operações*, representa uma abordagem colaborativa e cultural para o desenvolvimento de um software que busca integrar eficientemente as equipes de desenvolvimento (Dev) e operações (Ops). Essa prática visa quebrar as barreiras tradicionais entre essas duas áreas, promovendo a colaboração contínua e a entrega mais rápida e confiável de software.

Ao adotar o DevOps, as organizações buscam superar desafios históricos, como a falta de comunicação efetiva entre desenvolvimento e operações, demora na entrega de novas funcionalidades e dificuldades de identificação e correção rápida de problemas em produção. É por isso que áreas como a engenharia de software, embora possam parecer teóricas e não sejam especificamente conhecidas por suas contribuições diretas ao DevOps, fornecem uma base sólida, que pode ser aplicada e integrada, em abordagens como o DevOps, para melhorar a eficiência e a qualidade no desenvolvimento de software. Portanto, tenha em mente outros conhecimentos que você já adquiriu neste curso, pois eles serão importantes em nosso estudo!

O DevOps é quase um compêndio de várias técnicas e processos práticos que visam otimizar o trabalho na área de TI. É sobre todo esse mundo que



falaremos neste curso: sua história, personagens, técnicas e formas de implementação. Tenha em mente que o mercado de TI exige e exigirá cada vez mais conhecimento.

Figura 1 – O aquecimento do mercado no Brasil, motivado principalmente pela expansão de novas tecnologias

Aquecido no Brasil, mercado de TI indica aumento de vagas de trabalho

Órgão, no entanto, prevê dificuldades para preenchê-las pela falta de qualificação

Fonte: Senac Divinópolis, 2022.

Ou seja: existem, hoje, vagas e mais vagas na área, mas existe também a constante necessidade de estudo e crescimento. Se você gosta de novidades e de pesquisas, está no local e mercado certos. DevOps é tudo isso! É a soma de várias tecnologias históricas com as novas, em uma busca constante pela melhoria, no dia a dia em TI.

TEMA 1 – O QUE É DEVOPS E POR QUE ELE IMPORTA

DevOps é uma abordagem que integra o desenvolvimento de *software* (*Dev*) com as operações de TI (*Ops*) para melhorar continuamente a entrega de *software*. O objetivo do DevOps é reduzir as barreiras entre as equipes de desenvolvimento e operações, promovendo uma cultura de colaboração e automação.

1.1 O início do DevOps

Em 2008, Patrick Debois propôs discutir métodos para solucionar os conflitos entre desenvolvimento e operações de TI. Em uma palestra feita na Conferência de Agile, em Toronto, Debois externou acreditar que a infraestrutura era a fonte das ineficiências. No entanto, sua visão sobre desenvolvimento também era parcial. A virtude do seminário foi levantar a discussão sobre a necessidade de mudar métodos, para se homogeneizar o trabalho em TI (Kim et al., 2016).

O *agile* revolucionou as abordagens das metodologias tradicionais de desenvolvimento de *software*, como o modelo cascata, direcionando-as para um



ciclo contínuo de desenvolvimento. A expressão *DevOps* foi cunhada durante a Conferência Velocity da O'Reilly, em 2009, durante a apresentação de John Allspaw (Etsy.com) e Paul Hammond (Typekit). O propósito dessa conferência era reunir desenvolvedores (*Dev*) e administradores de infraestrutura de TI (*Ops*), promovendo uma integração contínua até a entrega do *software*. Um entusiasta significativo do tema, presente na conferência, foi o já mencionado Patrick Debois. Inspirado pela palestra mencionada, ele teve a brilhante ideia de criar um evento chamado *DevOpsDay*, cujo primeiro encontro ocorreu em Ghent, Bélgica, no final de 2009, estendendo-se por dois dias. Foi nesse evento que o conceito começou a se popularizar globalmente.

1.2 Definições e princípios do DevOps


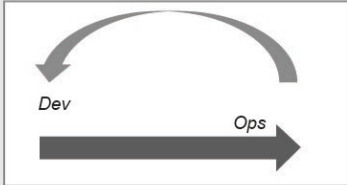
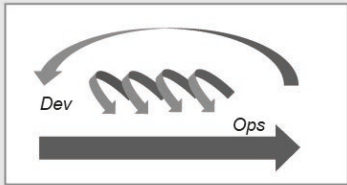
Segundo Antonio Muniz (2020), a primeira abordagem compreende princípios e práticas que favorecem o rápido desenvolvimento operacional, concentrando-se na execução de ações para acelerar o fluxo da esquerda para a direita, visando à redução do tempo de implementação. Os benefícios dessa abordagem incluem:

- Melhor visualização do trabalho
- Diminuição do tamanho dos lotes e dos intervalos
- Eliminação de desperdícios, com foco no cliente
- Integração da qualidade, desde a origem

Para ilustrá-lo, temos o que é descrito na Figura 2, na qual é demonstrado o fluxo esquerda-direita, de desenvolvimento para operações.



Figura 2 – Primeira, segunda e terceira abordagens de DevOps, considerando a aceleração de fluxo esquerda-direita

Objetivo da Primeira Maneira	Princípios e práticas
<p>Acelerar o fluxo dos desenvolvedores (esquerda) para operação e clientes (direita)</p> 	<ol style="list-style-type: none">1 Tornar o trabalho visível2 Reduzir o tamanho dos lotes e intervalos3 Aplicar a teoria das restrições e otimizar o fluxo4 Remover desperdícios e foco no cliente5 Reduzir o número de transferências (<i>hand-off</i>)6 Incorporar qualidade na origem7 Limitar o trabalho em andamento (WIP)8 Infraestrutura como código e <i>self-service</i>9 Integração, entrega e implantação contínua10 Testes automatizados e TDD11 Arquitetura e <i>releases</i> de baixo risco
Objetivo da Segunda Maneira	Princípios e práticas
<p>Rápido feedback em todos os estágios do fluxo de valor (direita para a esquerda)</p> 	<ol style="list-style-type: none">1 Ver problemas quando ocorrem ("<i>ir ap gembu</i>")2 Aglomerar quando o problema aparece (<i>Andon</i>)3 Qualidade próxima da fonte (menos aprovações)4 Telemetria <i>self-service</i> e irradiadores de informação disponíveis para todos5 Desenvolvimento por hipótese e testes A/B6 Equipes <i>Dev</i> e <i>Ops</i> compartilham o trabalho diário e plantões de suporte 24x77 Revisão de código usando as técnicas: programação em par, sobre os ombros, divulgação por e-mail, assistida por ferramentas
Objetivo da Terceira Maneira	Princípios e práticas
<p>Cultura de alta confiança que permite correr riscos e potencializar o aprendizado contínuo</p> 	<ol style="list-style-type: none">1 Cultura justa e segura para aprender e evoluir com os erros2 Injeção de falhas na produção para aumentar resiliência3 Converter descobertas locais em melhorias globais4 Reservar tempo para melhorar o trabalho diário (<i>Kata</i> e <i>Blitz</i> de melhoria)5 Reunião de <i>post mortem</i> sem culpa6 Instituir dias de jogos para ensaiar falhas7 Difundir conhecimento usando testes automatizados como documentação

Fonte: Muniz, 2020, p. 33-35.

A filosofia do DevOps é dividida igualmente entre automação e colaboração. No primeiro pilar, os benefícios mencionados podem ser resumidos como a qualidade originada, termo essencial nesse movimento e que é crucial para a entrega da experiência ao cliente. A segunda abordagem, vista na Figura 2, possibilita um fluxo contínuo e rápido de *feedback* do cliente para o desenvolvimento, em todas as etapas do ciclo de valor. Essa abordagem exige amplificação do *feedback* para evitar a recorrência de problemas. Entenda, por isso, que o processo de TI cria, nesse caso, o que chamamos de *pipeline* ou *cano*, ou seja, promove processamentos e entregas sucessivas.

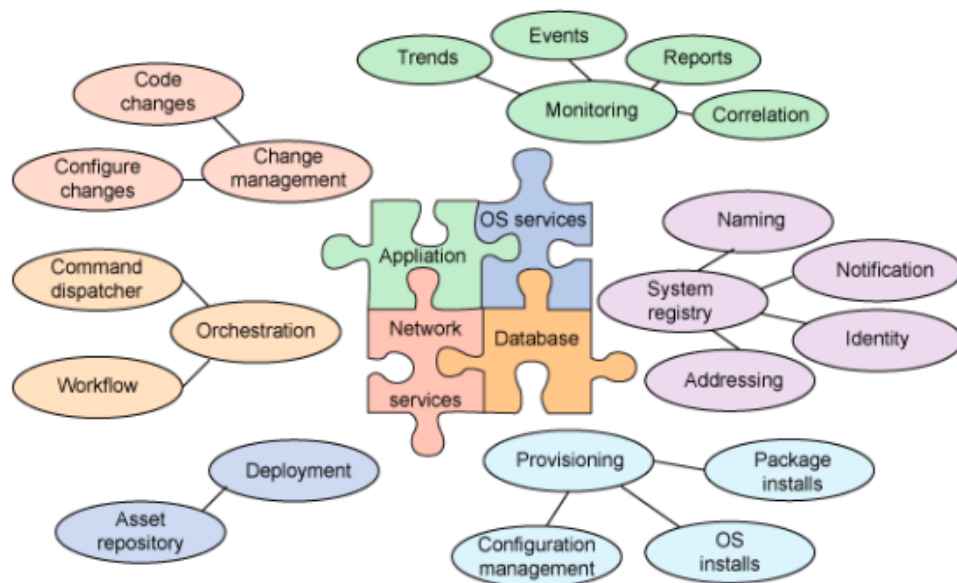
Alguns princípios fundamentais da segunda abordagem incluem:



- Telemetria
- Desenvolvimento baseado em hipóteses e testes A/B
- Programação em pares
- Programação sobre os ombros
- Qualidade próxima à fonte

De fato, de acordo com a Figura 2, são muitos os assuntos para serem tratados de forma aleatória, e um processo envolvendo tanto operações como desenvolvimento é necessário. E as operações são muitas, como pode ser visto na Figura 3, que ilustra um intrincado quebra-cabeças envolvendo diferentes operações de bancos, serviços, segurança, entre outras.

Figura 3 – Diversas frentes de trabalho para serem organizadas



Fonte: Definindo, 2013.

Finalmente, chegamos à terceira abordagem, descrita por Antonio Muniz (2020) como a criação de uma cultura de alta confiança, encorajando a tomada de riscos e promovendo a aprendizagem contínua. Isso permite a adoção de uma cultura de experimentação. Algumas práticas adotadas nessa abordagem incluem:

- Cultura organizacional baseada em confiança, desempenho e gestão sem culpa ou medo
- Institucionalização da melhoria do trabalho diário
- Transformação de descobertas locais em melhorias globais



- Estabelecimento de padrões de resiliência no trabalho diário
- Fomento a uma cultura de aprendizado

Resumindo, temos três abordagens fundamentais para a implementação eficaz de práticas DevOps. A primeira abordagem, conforme delineada por Antonio Muniz (2020), enfatiza a execução de ações direcionadas à aceleração do fluxo operacional, reduzindo o tempo de implementação e eliminando desperdícios. Por sua vez, a segunda abordagem destaca a importância do *feedback* contínuo do cliente ao longo de todas as etapas do ciclo de valor, com práticas colaborativas e ágeis. Por fim, a terceira abordagem busca estabelecer uma cultura organizacional baseada em confiança e aprendizado contínuo, incentivando a experimentação e a melhoria contínua do trabalho diário.

A integração entre operações e desenvolvimento, conforme evidenciado, revela-se um desafio complexo, exigindo não apenas a adoção de ferramentas e práticas específicas, mas também uma mudança cultural profunda, dentro das organizações. As Figuras 2 e 3 ilustram essa complexidade, destacando a necessidade de uma abordagem holística e integrada para se alcançar os objetivos propostos. Em suma, as abordagens discutidas fornecem um quadro abrangente para a implementação bem-sucedida de DevOps, destacando a importância da automação, colaboração e cultura organizacional para se alcançar a excelência operacional e a satisfação do cliente.

Agora vamos abordar o conceito de DevOps não de forma teórica, mas aplicada, mais para uma abordagem prática e cultural do que para um conjunto de definições acadêmicas com autores específicos. Alguns especialistas e líderes de pensamento no campo contribuíram para moldar e promover os princípios do DevOps. O DevOps possibilita que funções previamente segregadas, como desenvolvimento, operações de TI, engenharia da qualidade e segurança, trabalhem de maneira sincronizada e colaborativa para criar produtos mais robustos e confiáveis. Ao abraçar uma mentalidade e ferramentas de DevOps, as equipes adquirem a habilidade de responder de forma mais eficaz às demandas dos clientes, aumentar a confiança nos aplicativos que desenvolvem e atingir objetivos empresariais com maior agilidade (Kim et al., 2016).

Vamos detalhar com cuidado, mais para a frente; mas, tecnicamente existe, uma série de princípios do DevOps, que a seguir resumimos (O que é, [S.d.]):



- **Cultura de colaboração:** promover uma cultura que valorize a colaboração e a comunicação eficaz entre equipes de desenvolvimento e operações.
- **Automação:** buscar a automação de processos, desde o desenvolvimento até a produção, para aumentar eficiência, reduzir erros e possibilitar entregas mais rápidas.
- **Entrega contínua:** adotar a prática de entrega contínua, permitindo a liberação rápida e frequente de *software* de alta qualidade.
- **Monitoramento e *feedback*:** implementar sistemas de monitoramento e *feedback* para garantir visibilidade em tempo real do desempenho do aplicativo e identificar áreas de melhoria.
- **Infraestrutura como código (IaC):** tratar a infraestrutura como código, possibilitando a automação e o versionamento de ambientes, aumentando a consistência e facilitando a replicação.
- **Gestão de configuração:** utilizar ferramentas de gestão de configuração para controlar e gerenciar alterações na infraestrutura e no código de maneira eficiente.
- **Segurança integrada:** integrar práticas de segurança desde as fases iniciais do desenvolvimento, garantindo que a segurança seja uma consideração contínua.
- **Melhoria contínua:** buscar, constantemente, aprimorar processos e práticas, aprendendo com os sucessos e falhas e promovendo uma cultura de melhoria contínua.
- **Resiliência:** projetar sistemas e práticas para serem resilientes a falhas, minimizando impactos e permitindo rápida recuperação.
- **Desenvolvimento e operações como equipe única:** fomentar a ideia de que desenvolvimento e operações não são silos separados, mas trabalham como uma equipe integrada para alcançar objetivos comuns.

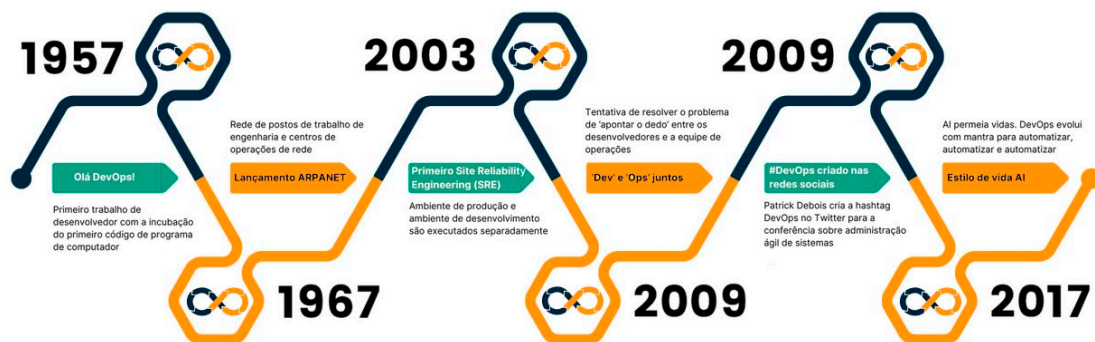
1.3 História e contextualização do DevOps

Apesar de termos mencionado o contexto de criação do DevOps, a prática não pertence somente a um único autor. Ela evoluiu e atualmente é muito vasta, é mais uma abordagem prática e cultural do que um conjunto de definições acadêmicas com autores específicos. No entanto, alguns especialistas e líderes



de pensamento no campo contribuíram para moldar uma abordagem para o desenvolvimento de *software* que busca integrar mais estreitamente as equipes de desenvolvimento e operações para melhorar a colaboração e eficiência em todo o ciclo de vida do *software*. A história e a contextualização do DevOps podem ser resumidas em alguns marcos significativos (Enap, 2020), conforme Figura 4.

Figura 4 – Histórico do DevOps



Fonte: Ferste, 2024.

Entendendo a Figura 4, durante a fase 1, que se estende de 1957 a 2003, destacam-se marcos importantes no desenvolvimento da automação e no estabelecimento de metodologias de desenvolvimento de *software*, como o surgimento do primeiro compilador de linguagem de programação, em 1957, e a adoção do modelo *waterfall*, em 1967. Na fase 2, de 2003 a 2009, ocorre uma transição significativa com o início do movimento *agile*, enfatizando entregas frequentes e *feedback* constante, além do surgimento do termo *DevOps* pela primeira vez em 2009, marcando a combinação entre práticas de desenvolvimento e operações. Finalmente, na fase 3, de 2009 a 2017, há um aumento significativo na adoção de práticas DevOps, como a integração contínua e a entrega contínua (CI/CD), culminando com 2017, quando o DevOps se torna uma prática *mainstream*, com empresas de todos os tamanhos buscando seus benefícios.

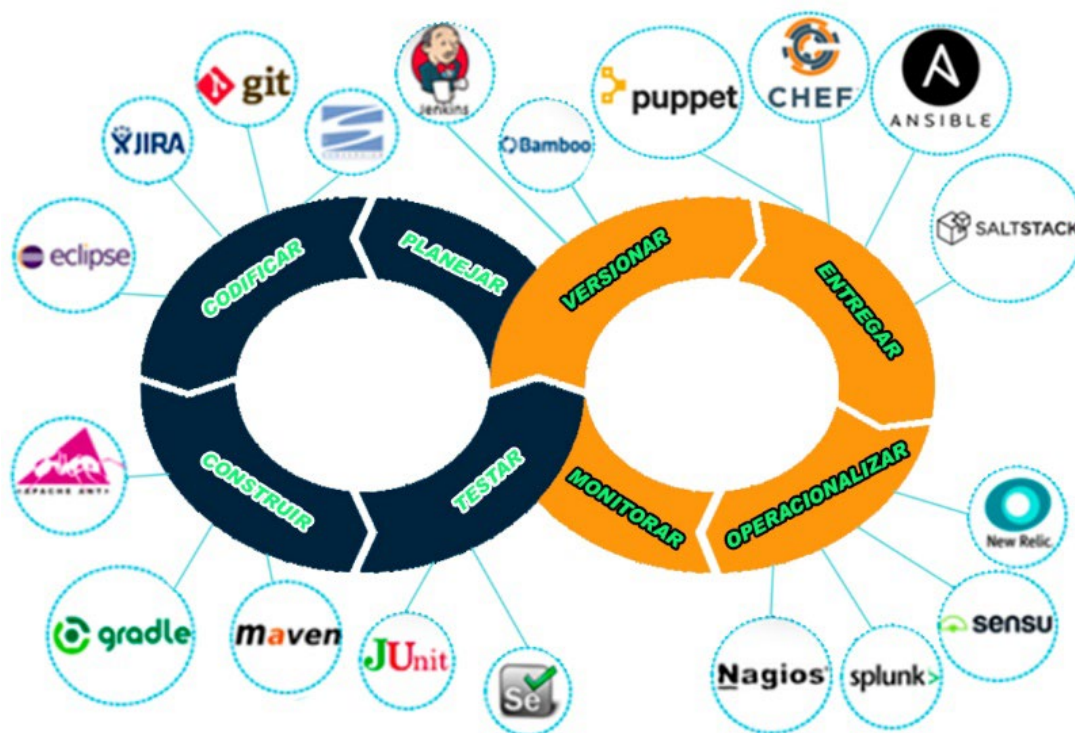


1.4 Benefícios de DevOps

A integração entre duas áreas cruciais em uma empresa, *development* (desenvolvimento) e *operations* (operações), traz consigo uma série de vantagens, conforme detalhado a seguir.

- **Entregas rápidas:** aumento na frequência e ritmo de lançamentos, possibilitando inovação constante e aprimoramento dos produtos. Essa agilidade proporciona uma vantagem competitiva para o negócio.
- **Confiabilidade:** maior segurança nas entregas devido à qualidade aprimorada nas atualizações de aplicativos e alterações de infraestrutura. Isso não apenas garante agilidade, mas também resulta em experiências positivas para os usuários finais.
- **Velocidade:** facilitação da oferta de inovações aos clientes, otimização da eficiência na obtenção de resultados e adaptação dinâmica às demandas do mercado.
- **Colaboração:** promoção da eficiência por meio de equipes que trabalham em colaboração e compartilham responsabilidades, fomentando uma abordagem mais integrada.
- **Escala:** realização de processos de infraestrutura e desenvolvimento em larga escala. Isso resulta em um gerenciamento mais eficiente de sistemas complexos, reduzindo riscos operacionais.
- **Segurança:** implementação do modelo DevOps sem comprometer a segurança, graças a **políticas** automáticas de conformidade, controles rigorosos e práticas de gerenciamento de configuração. Isso assegura que a segurança seja parte integrante de todo o ciclo de vida do desenvolvimento e operações.

Figura 5 – Grande parte do DevOps está na automação do ciclo



Fonte: Ferste, 2024.

Um dos princípios fundamentais do DevOps é o investimento em automação, como é visto na Figura 5. A automação possibilita a execução mais rápida de tarefas ou processos, reduzindo significativamente a probabilidade de erros humanos. Diversas ferramentas são empregadas para atingir esse objetivo:

- **Vagrant:** essa ferramenta facilita a construção de ambientes de desenvolvimento virtualizados completos. Seu fluxo de trabalho é simples e focado em automação.
- **Docker:** como uma plataforma aberta, o Docker auxilia desenvolvedores e administradores de sistemas na criação e execução de aplicações distribuídas.
- **Puppet:** trata-se de uma ferramenta de código aberto voltada para o gerenciamento de configuração. Centraliza a configuração e a distribui para várias máquinas ou servidores na rede.
- **Chef:** o Chef possibilita a automação do processo de construção, implantação e gerenciamento da infraestrutura, tornando-a versionável e testável.



- **Composer:** esse é um gerenciador de dependências de aplicação, facilitando a manutenção e inclusão de novos pacotes ou bibliotecas necessárias na aplicação.
- **New Relic:** trata-se de uma ferramenta de monitoramento de aplicação que permite a análise profunda do desempenho da aplicação. Ajuda os desenvolvedores a compreenderem as nuances dos dados históricos por meio da coleta, armazenamento e análise dessas informações. Vale ressaltar que existem diversas ferramentas disponíveis, e a escolha de uma delas deve ser feita com base nas necessidades específicas a cada caso.

TEMA 2 – DEVOPS: EVOLUÇÃO E APLICAÇÕES PRÁTICAS

As aplicações práticas do DevOps abrangem uma variedade de atividades e processos que visam melhorar a colaboração, eficiência e qualidade ao longo do ciclo de vida do desenvolvimento e operações de *software*. A seguir estão algumas aplicações práticas comuns em DevOps. A implementação bem-sucedida dessas práticas no contexto do DevOps contribui para uma entrega de *software* mais eficiente, confiável e alinhada aos objetivos de negócios.

2.1 Problemas na comunicação

Antes do DevOps, as equipes de desenvolvimento e operações frequentemente operavam em silos separados, isso resultando em problemas de comunicação, atrasos e falta de alinhamento. Os problemas de comunicação em TI podem ser desafios significativos que afetam a eficiência, a colaboração e, consequentemente, o desempenho global de uma equipe ou departamento.

Alguns dos problemas comuns de comunicação em TI (O que é, [S.d.]) consistem em:

- **Barreiras de silos:** divisões tradicionais entre equipes, como desenvolvimento, operações, suporte e segurança, podem criar silos de informação. A falta de comunicação entre esses silos pode levar a mal-entendidos, duplicação de esforços e atrasos.
- **Linguagem técnica complexa:** a TI frequentemente utiliza terminologia técnica complexa. Quando essa linguagem não é traduzida de maneira eficaz para as partes interessadas não técnicas, ocorrem falhas de



comunicação, dificultando a compreensão e a tomada de decisões informadas.

- **Falta de documentação adequada:** a ausência de documentação clara sobre códigos, configurações de sistemas e processos pode levar a mal-entendidos e dificultar a colaboração. A documentação inadequada também pode resultar em retrabalho e perda de tempo.
- **Má gestão de expectativas:** falhas na comunicação de prazos, metas e requisitos podem levar a expectativas não correspondidas. Isso pode causar frustração nas equipes e entre os *stakeholders*, resultando em desconfiança e descontentamento.
- **Falta de transparência:** quando informações importantes não são compartilhadas de maneira transparente, podem ocorrer problemas. Isso inclui informações sobre mudanças nos requisitos, problemas de segurança e atualizações no *status* de projetos.
- **Comunicação virtual ineficaz:** com equipes distribuídas geograficamente ou trabalhando remotamente, a comunicação virtual se torna crucial. Problemas como falta de sincronia de fusos horários, mal-entendidos em mensagens escritas e falta de interações cara a cara podem surgir.
- **Resistência à mudança:** a introdução de novas tecnologias, processos ou metodologias frequentemente encontra resistência. A falta de comunicação eficaz sobre os motivos para a mudança, os seus benefícios e o suporte disponível podem resultar em resistência e falta de colaboração.
- **Feedback inadequado:** a comunicação bidirecional é essencial. A falta de *feedback* construtivo e a recusa em aceitar *feedback* podem levar a erros persistentes e a uma falta de melhoria contínua.
- **Falta de compreensão dos objetivos de negócios:** quando as equipes de TI não compreendem totalmente os objetivos e as prioridades do negócio, podem surgir decisões e implementações que não atendem às necessidades mais amplas da organização.
- **Desalinhamento entre desenvolvimento e operações:** no contexto DevOps, o desalinhamento entre equipes de desenvolvimento e operações pode levar a atrasos na entrega, instabilidade em ambientes de produção e problemas de escalabilidade.



Resolver esses problemas de comunicação em TI requer esforços contínuos para promover uma cultura de comunicação aberta, investir em ferramentas e processos eficazes e incentivar a colaboração interdisciplinar. Uma comunicação eficaz é fundamental para o sucesso de projetos e operações de TI.

2.2 Necessidade de agilidade e eficiência

Segundo Pressman e Maxim (2014), a necessidade de agilidade e eficiência em TI é impulsionada por diversas demandas e desafios enfrentados pelas organizações na era digital. Aqui estão alguns postos-chaves que destacam essa necessidade:

- **Rápida evolução tecnológica:** a tecnologia está em constante evolução, e as organizações precisam se adaptar rapidamente para aproveitar as inovações. A agilidade permite que as equipes de TI respondam eficientemente às mudanças e adotem novas tecnologias de maneira oportuna.
- **Ciclos de vida de produtos mais curtos:** a pressão por lançar produtos e serviços mais rapidamente requer práticas ágeis para acelerar o seu desenvolvimento e entrega, garantindo que as organizações permaneçam competitivas, em mercados dinâmicos.
- **Maior demanda por experiência do usuário:** a expectativa dos usuários por experiências digitais excepcionais está em ascensão. A agilidade permite que as equipes de TI ajustem rapidamente as aplicações e serviços para atender às expectativas dos usuários e proporcionar uma experiência mais satisfatória.
- **Ambientes de negócios voláteis:** a agilidade em TI permite que as organizações se adaptem rapidamente a mudanças nas condições de mercado, regulamentações e fatores externos, garantindo uma resposta ágil a eventos imprevistos.
- **Mudança cultural para DevOps:** a adoção de práticas DevOps visa melhorar a colaboração entre equipes de desenvolvimento e operações, acelerando os ciclos de desenvolvimento e aumentando a eficiência na entrega e manutenção de software.



- **Necessidade de escalabilidade:** empresas em crescimento precisam escalar suas operações de TI de maneira eficiente. A agilidade e a eficiência são essenciais para garantir que os sistemas e processos possam lidar com o aumento da demanda.
- **Gestão de custos e recursos:** a eficiência em TI envolve otimizar o uso de recursos, reduzir custos operacionais e garantir que os investimentos em tecnologia gerem valor tangível para o negócio.
- **Resposta a mudanças de requisitos:** a natureza dinâmica dos requisitos de negócios exige uma capacidade ágil para ajustar prioridades, realizar mudanças no escopo do projeto e fornecer soluções alinhadas às necessidades emergentes.
- **Satisfação do cliente:** a agilidade e a eficiência em TI contribuem diretamente para a satisfação do cliente, garantindo entregas rápidas, confiáveis e alinhadas às expectativas.

Em resumo, agilidade e eficiência, em TI, são essenciais para que as organizações enfrentem os desafios dinâmicos do ambiente digital, impulsionem a inovação, alcancem seus objetivos estratégicos e forneçam valor aos clientes, de maneira contínua.

2.3 Cultura e colaboração

A cultura de colaboração em TI é essencial para promover um ambiente de trabalho coeso e eficaz, em que as equipes interagem de maneira integrada para alcançar objetivos comuns. Essa abordagem transcende barreiras organizacionais, incentivando a comunicação aberta, o compartilhamento de conhecimento e a formação de equipes multifuncionais. No contexto da TI, a colaboração frequentemente envolve a integração de desenvolvimento e operações, em que as equipes colaboram para otimizar processos e acelerar o ciclo de vida de desenvolvimento. Essa abordagem também incentiva a transparência na comunicação, permitindo que informações sobre projetos e desafios sejam compartilhadas de maneira ampla (TDD, 2023).

Uma cultura de colaboração valoriza a inovação, pois reúne diversas perspectivas e habilidades para resolver problemas de maneira criativa. Além disso, promove a autonomia responsável, em que os membros da equipe têm a liberdade de tomar decisões com base nas informações disponíveis. Líderes



desempenham um papel crucial ao facilitarem a construção desse ambiente, estimulando um espírito de equipe, reconhecendo e celebrando conquistas coletivas e fomentando uma mentalidade de *feedback* construtivo.

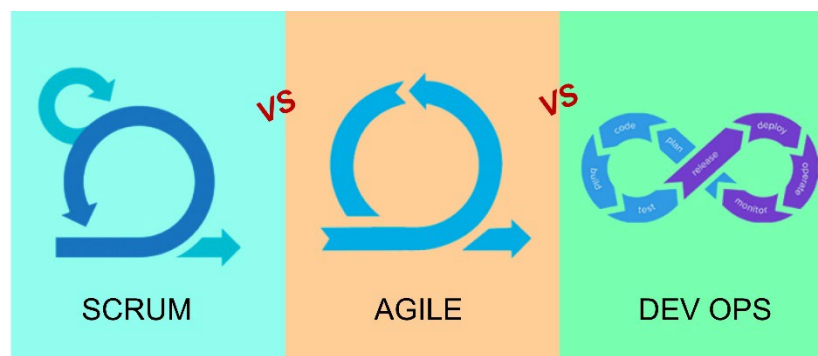
Em resumo, a cultura de colaboração em TI não apenas fortalece as relações internas, mas também impulsiona a inovação, a eficiência operacional e a capacidade de adaptação a mudanças. Essa abordagem é vital para se enfrentar os desafios dinâmicos do setor de tecnologia e alcançar resultados excepcionais.

TEMA 3 – DESENVOLVIMENTO ÁGIL E DEVOPS

O termo *desenvolvimento ágil* ou *agilismo* refere-se a uma abordagem flexível e colaborativa para o desenvolvimento de *software*. Essa filosofia, baseada no *Manifesto ágil*, busca superar as limitações das abordagens tradicionais, promovendo a adaptabilidade, a entrega contínua e a satisfação do cliente. O *Manifesto ágil* estabelece valores fundamentais, como priorizar indivíduos e interações sobre processos e ferramentas, entregar *software* funcional mais que documentação abrangente, colaborar com o cliente em vez de negociar contratos e responder a mudanças em vez de seguir um plano. Os princípios do *Manifesto ágil* orientam a aplicação prática desses valores, promovendo a entrega frequente de *software* funcional, adaptabilidade a mudanças nos requisitos, colaboração diária entre desenvolvedores e partes interessadas e construção de projetos por indivíduos motivados.

Diversas metodologias ágeis, como *scrum*, *extreme programming* (programação extrema – XP) e *kanban*, foram desenvolvidas para implementar esses princípios, como visto na Figura 6. Todas essas metodologias são diferentes, mas parecidas.

Figura 6 – Diferenças e semelhanças entre *scrum agile* e DevOps





O desenvolvimento ágil oferece benefícios significativos, como adaptabilidade eficiente, entrega contínua de incrementos de valor, satisfação do cliente, colaboração efetiva e promoção de melhoria contínua nas práticas de desenvolvimento. Essa abordagem é amplamente adotada na indústria de *software* devido à sua capacidade de oferecer flexibilidade, respostas rápidas e resultados alinhados às necessidades dos clientes.

3.1 Agilismo e DevOps

Nos primórdios da TI, um muro alto separava desenvolvimento e operações. Desenvolvedores concentravam-se na criação de códigos, enquanto equipes de operações lidavam com implantação, infraestrutura e manutenção. Essa divisão frequentemente resultava em ciclos de desenvolvimento demorados, lançamentos pouco frequentes e desafios significativos de comunicação.

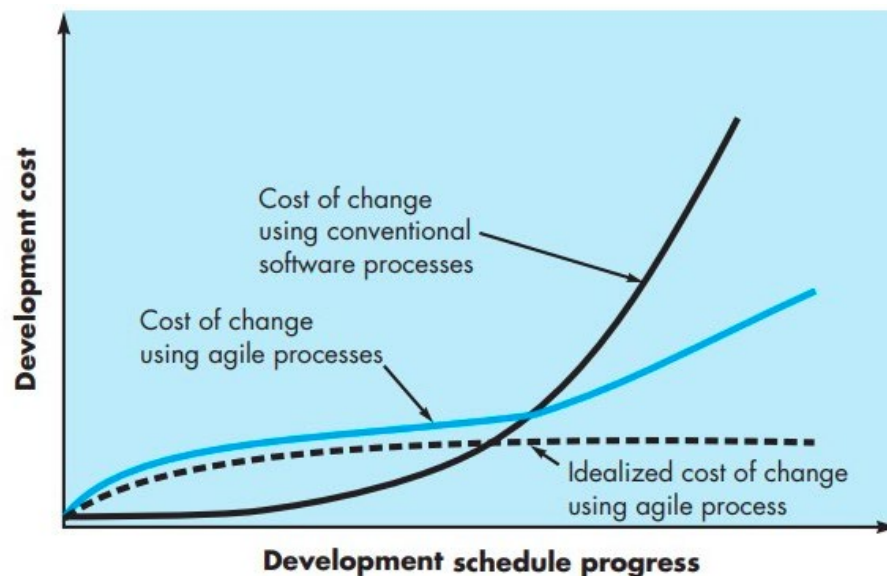
O movimento ágil, com sua ênfase em ciclos de desenvolvimento mais curtos (*sprints*) e *feedback* contínuo, foi um dos primeiros a desafiar essa dinâmica. No entanto, mesmo com o *Manifesto ágil*, uma lacuna persistia entre desenvolver *softwares* e implementá-los de maneira eficaz. A grande virada ocorreu por volta de 2008-2009, durante uma conferência em Ghent, na Bélgica, em que o termo *DevOps* foi verdadeiramente concebido. A ideia era promover uma colaboração mais estreita entre desenvolvimento e operações, integrando-os, em termos de trabalho, em uma equipe única, formada quase como que por amigos de infância ou, pelo menos, que deveriam ser unidos, certo? O objetivo disso era combinar a agilidade do desenvolvimento com a estabilidade e a eficiência das operações.

Inicialmente adotada principalmente por *startups* e empresas de tecnologia, a cultura DevOps rapidamente ganhou aceitação. Com o crescimento da computação em nuvem e a necessidade de se lançar atualizações frequentes para atender às expectativas dos usuários e às mudanças rápidas do mercado, o DevOps tornou-se praticamente uma necessidade. Atualmente, grandes corporações e até mesmo indústrias tradicionais adotam práticas DevOps para aprimorar sua eficiência, qualidade e velocidade no mercado.

As organizações estão em busca de inovação no desenvolvimento de seus *softwares*, e uma das chaves para elas alcançarem o sucesso reside na adoção das práticas ágil e DevOps. Essas são tendências globais que passam

por ajustes constantes, visando a seu aprimoramento contínuo. A engenharia de *software* ágil surgiu devido à sua rapidez na implementação e ao seu impacto positivo na satisfação do cliente. A aplicação de planejamento é mais eficiente, e há uma facilidade notável no desenvolvimento geral, tornando-se ela, em muitos casos, uma alternativa superior ao método tradicional (Figura 7).

Figura 7 – Custos em função do tempo de desenvolvimento



Fonte: Pressman; Maxim, 2014.

3.2 Extreme programming (XP)

Uma metodologia amplamente adotada nas empresas atualmente é a XP. Ela inicia seu processo com um planejamento detalhado, em que a coleta de requisitos é prioritária para se compreender as necessidades comerciais do *software*, seus recursos e funcionalidades. A história é então escrita pelo cliente, que lhe atribui valores de prioridade com base no impacto geral no negócio, recursos e funções.

Em seguida, os membros da equipe de XP estimam o custo médio de desenvolvimento, em semanas. Caso a estimativa para a implementação da história seja superior a três semanas, o cliente é convidado a subdividi-la em tarefas menores. A equipe de XP realiza análises de estimativas sucessivas. Os desenvolvedores colaboram diretamente com os clientes para definir os próximos lançamentos de recursos e funcionalidades do *software*, sempre incluindo a data de entrega e os detalhes do projeto.



As histórias são desenvolvidas de acordo com os princípios mostrados na Figura 8. Além disso:

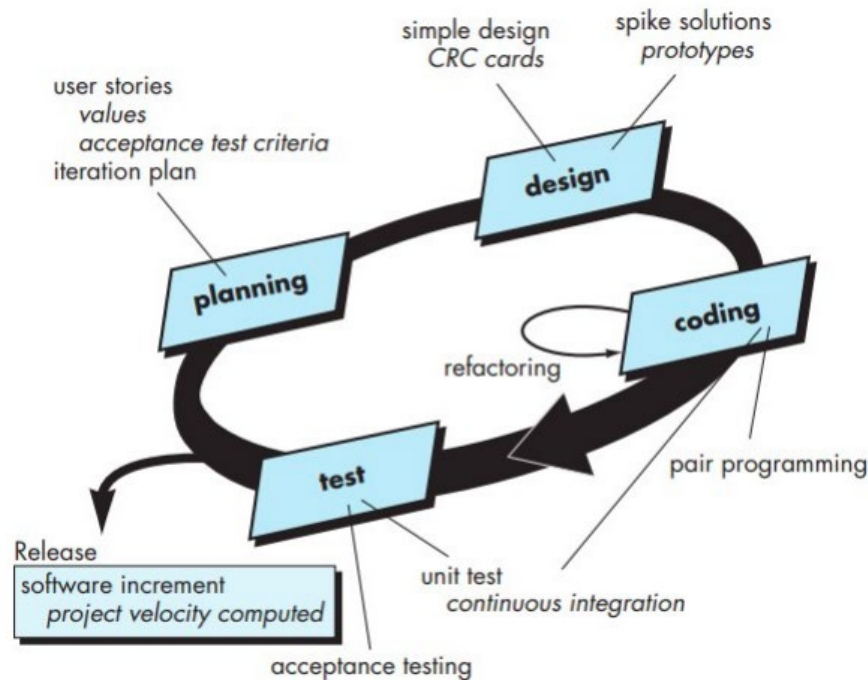
- Todas as histórias são implementadas em algumas semanas.
- As histórias de maior valor têm prioridade e são implementadas primeiro.
- Histórias mais arriscadas são tratadas em separado, no cronograma.

A velocidade da implementação do projeto é calculada com base no número de histórias entregues no primeiro lançamento, e o planejamento é ajustado conforme o desenvolvimento contínuo. Durante essa fase, o cliente pode adicionar ou retirar histórias.

A fase de desenho segue rigorosamente o princípio *keep it simple, stupid* (Kiss), priorizando representações menos complexas. A XP incentiva o uso de cartões de classe, responsabilidade, colaboração (CRC) para promover o pensamento orientado a objetos. Quando um problema de *design* é desafiador, a XP recomenda se criar imediatamente um protótipo operacional dessa parte do *design* para se reduzir riscos sem prejudicar as validações e estimativas originais.

A refatoração é uma parte crucial do desenvolvimento, incentivada pela equipe para melhorar a estrutura interna do código, sem alterar seu comportamento. Esse processo é contínuo, ao longo do desenvolvimento do sistema. Os testes unitários são realizados para cada história desenvolvida, proporcionando *feedback* imediato aos desenvolvedores, após o sucesso dos testes. A programação em pares é um conceito importante na XP e ocorre quando duas pessoas trabalham juntas em uma estação de trabalho, proporcionando uma solução de problemas em tempo real e garantindo a qualidade do código, já que é ele revisado continuamente durante a criação do programa.

Figura 8 – Processo de desenvolvimento da metodologia XP



Fonte: Pressman; Maxim, 2014.

TEMA 4 – A CULTURA DEVOPS: PRINCÍPIOS E VALORES

A cultura DevOps é fundamental para o sucesso da implementação de práticas e ferramentas associadas ao DevOps. Ela enfatiza a colaboração, a comunicação eficaz e a busca constante pela melhoria na entrega de *software* (O que é, [S.d.]).

Consistem em princípios da cultura DevOps:

a. Integração contínua (CI)

- O que é: automação da integração de código de diferentes contribuidores em um único projeto, várias vezes ao dia, garantindo que as alterações sejam integradas e testadas de maneira contínua.
- Benefícios: redução de conflitos de código, identificação rápida de erros e melhoria da qualidade do software.

b. Entrega contínua (CD)

- O que é: automação do processo de liberação de software para ambientes de teste e produção, permitindo entregas frequentes e confiáveis.
- Benefícios: redução do tempo entre o desenvolvimento e a disponibilidade do produto, facilitando a inovação e a adaptação rápida às mudanças.



c. **Automação de infraestrutura**

- O que é: utilização de ferramentas para automatizar a configuração e o provisionamento de ambientes de infraestrutura.
- Benefícios: maior consistência na implantação de ambientes, redução de erros manuais e agilidade na escalabilidade.

d. **Monitoramento contínuo**

- O que é: implementação de soluções de monitoramento para acompanhar o desempenho do software em tempo real.
- Benefícios: detecção precoce de problemas, feedback rápido sobre o desempenho do aplicativo e melhoria contínua.

e. **Testes automatizados**

- O que é: automação de testes de unidade, integração e aceitação para garantir a qualidade do código.
- Benefícios: identificação precoce de *bugs*, garantia de consistência nos testes e aceleração do ciclo de desenvolvimento.

f. **Gerenciamento de configuração**

- O que é: controle e automação da configuração de software e infraestrutura.
- Benefícios: maior consistência entre ambientes, fácil reprodução de configurações e redução de falhas relacionadas à configuração.

g. **Colaboração entre equipes**

- O que é: promoção de uma cultura de colaboração entre equipes de desenvolvimento, operações e outras partes interessadas.
- Benefícios: compartilhamento eficiente de conhecimento, redução de conflitos e alinhamento com objetivos empresariais.

h. **Segurança DevSecOps**

- O que é: integração de práticas de segurança em todas as etapas do ciclo de vida do desenvolvimento.
- Benefícios: identificação precoce de vulnerabilidades, conformidade com padrões de segurança e redução de riscos.



TEMA 5 – CULTURA E PRÁTICAS DEVOPS NA VIDA REAL

Existem vários casos famosos de implementações bem-sucedidas de práticas DevOps em diversas organizações. A seguir analisaremos alguns exemplos notáveis, nesse sentido.

5.1 Amazon

Embora a Amazon seja conhecida por sua abordagem inovadora e bem-sucedida em práticas DevOps, é importante observar que a empresa não possui um **caso DevOps** específico ou uma única história que detalhe sua transformação. A cultura e as práticas DevOps na Amazon foram gradualmente incorporadas ao longo do tempo, refletindo uma evolução contínua em resposta às demandas do mercado e aos desafios operacionais.

5.1.1 História e evolução

A história e a evolução das práticas DevOps na Amazon perpassam:

- **Origens ágeis:** a Amazon começou a incorporar práticas ágeis em seu desenvolvimento de software no início dos anos 2000. Essa abordagem mais ágil foi uma resposta aos desafios percebidos nos ciclos de desenvolvimento mais longos e nas lacunas de comunicação entre desenvolvimento e operações.
- **Adoção progressiva de práticas DevOps:** ao longo dos anos, a Amazon adotou gradualmente práticas e princípios DevOps. Isso incluiu a implementação de entrega contínua, integração contínua, automação extensiva e uma forte ênfase na colaboração entre desenvolvimento e operações.
- **Cultura de inovação:** a cultura da Amazon, centrada na inovação e na busca incessante pela excelência do atendimento ao cliente, desempenhou um papel crucial na adoção de práticas DevOps. A empresa sempre esteve disposta a experimentar novas abordagens e tecnologias para melhorar a eficiência e a confiabilidade de seus serviços.
- **Pessoas notáveis: Jeff Bezos**, fundador e ex-diretor-executivo (ex-CEO) da Amazon, desempenhou um papel fundamental na criação de uma cultura de inovação e foco no cliente, na empresa. Sua liderança influente



ajudou a moldar a mentalidade da Amazon em direção à adoção de práticas ágeis e DevOps. Werner Vogels, diretor de tecnologia (CTO) da Amazon, também teve um papel significativo na transformação tecnológica da empresa. Sua liderança na equipe técnica contribuiu para a implementação bem-sucedida de práticas DevOps e a criação de uma infraestrutura escalável, na Amazon Web Services (AWS).

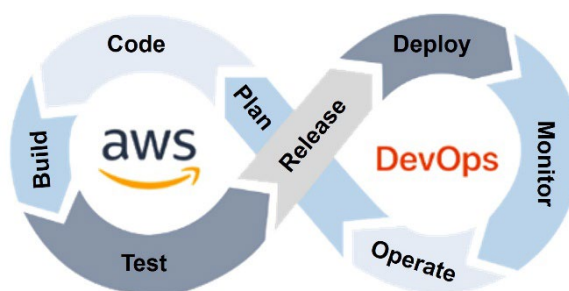
- **Equipes de engenharia e desenvolvimento:** as equipes de engenharia e desenvolvimento da Amazon, compostas por profissionais altamente qualificados, foram fundamentais na implementação eficaz de práticas DevOps. A colaboração próxima e a responsabilidade compartilhada entre essas equipes foram aspectos cruciais de sucesso.

5.1.2 Resultados e impactos

O alcance de resultados e impactos nas práticas DevOps na Amazon contemplam:

- **Inovação contínua:** a Amazon, por meio de suas práticas DevOps, continuou a inovar e lançar novos serviços, mantendo-se na vanguarda da indústria de tecnologia.
- **Eficiência operacional:** a implementação de práticas DevOps contribuiu para melhorias significativas na eficiência operacional da Amazon, permitindo ciclos de desenvolvimento mais curtos e entregas mais rápidas de produtos e serviços.
- **Liderança na nuvem:** a AWS, a divisão de serviços em nuvem da Amazon, tornou-se uma líder global devido à sua capacidade de fornecer serviços escaláveis e confiáveis, alimentada por práticas DevOps.

Figura 9 – Uma nova versão de DevOps adaptada pela AWS



Fonte: Ferste, 2024.



Embora não haja uma única história ou *case* específico da Amazon em relação a DevOps, a evolução e o sucesso da empresa no campo refletem uma adoção progressiva de práticas ágeis e DevOps em seu DNA organizacional. Isso destaca a importância contínua de uma abordagem flexível e adaptável para enfrentar os desafios de um mercado em constante mudança.

5.2 Netflix

Em 2008, a Netflix sofreu uma interrupção de serviço de três dias, devido a uma corrupção no seu banco de dados. Essa interrupção afetou um terço dos seus 8,4 milhões de assinantes, na época. O incidente levou a Netflix a tomar a decisão de migrar para a nuvem e realizar uma reforma completa em sua infraestrutura. A Netflix, então, escolheu a AWS como sua parceira de nuvem e levou quase sete anos para concluir a migração. A empresa não se limitou a simplesmente transferir seus sistemas para a nuvem, mas sim a reescrevê-los completamente para torná-los verdadeiramente nativos da nuvem. Essa transformação foi fundamental para a Netflix mudar a forma como operava. Na época, o então vice-presidente de engenharia de nuvem e plataforma da Netflix, Yury Izrailevsky, explicou que: “Percebemos que precisávamos nos afastar de pontos únicos de falha dimensionados verticalmente, como bancos de dados relacionais em nosso data center, para sistemas distribuídos, altamente confiáveis e horizontalmente escaláveis na nuvem”.

Como parte dessa transformação, a Netflix converteu seu aplicativo monolítico Java, baseado em *data center*, em uma arquitetura de microsserviços Java, baseada em nuvem. Essa mudança ofereceu várias vantagens à empresa, incluindo:

- Adoção de um modelo de dados desnormalizado, utilizando bancos de dados NoSQL.
- Liberdade para as equipes da Netflix se unirem de forma mais flexível.
- Capacidade para as equipes construírem e promoverem mudanças na velocidade com a qual se sentiam confortáveis.
- Implementação de uma coordenação de liberação centralizada.
- Abandono de ciclos de provisionamento de *hardware* de várias semanas, em favor da entrega contínua.
- Tomada de decisões independentes pelas equipes de engenharia, mediante uso de ferramentas de autoatendimento.

Figura 10 – DevOps na Netflix



Fonte: Dahibhate, 2023.

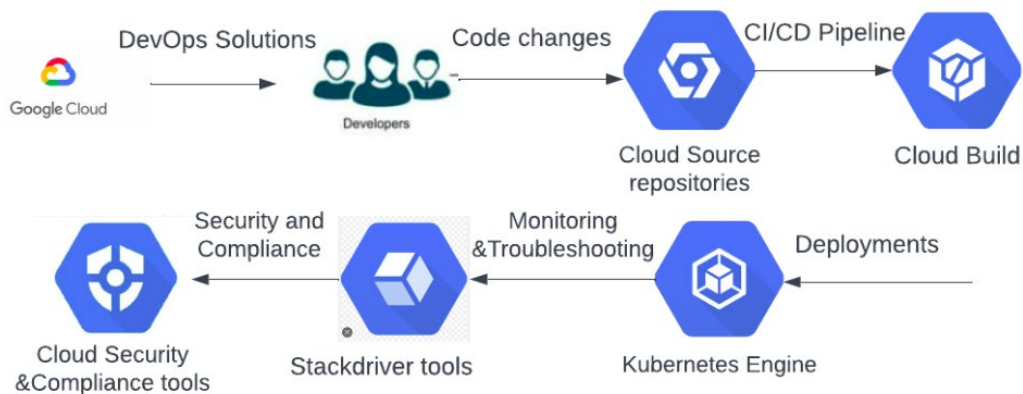
Como resultado dessa transformação, a Netflix acelerou a sua capacidade de inovação e adotou a cultura DevOps. A empresa também conquistou oito vezes mais assinantes do que em 2008, e as horas mensais de *streaming* da Netflix cresceram mil vezes, de dezembro de 2007 a dezembro de 2015.

5.3 Google

O Google, conhecido por sua abordagem inovadora, implementa práticas DevOps para suportar a implantação eficiente e escalável de seus serviços. A empresa usa automação extensiva e colaboração entre equipes para garantir a confiabilidade e a entrega contínua de seus produtos e serviços.



Figura 11 – A adoção do fluxo DevOps pelo Google



Fonte: Business, [S.d.].

5.4 Facebook

O Facebook adotou práticas DevOps para lidar com um ambiente de desenvolvimento altamente dinâmico. A empresa enfatiza a automação, os testes contínuos e a integração para garantir uma rápida entrega de novos recursos e melhorias, em suas práticas.

5.5 Outros exemplos

Existem vários outros exemplos. O Twitter implementou DevOps para melhorar a colaboração entre desenvolvedores e operações. Isso resultou em uma entrega mais rápida de novos recursos e maior estabilidade da plataforma. A Microsoft passou por uma transformação significativa para adotar práticas DevOps em suas operações e desenvolvimento de *software*. Ela integra CI/CD extensivamente e promove a colaboração entre equipes para desenvolver e entregar produtos mais rapidamente.

Esses exemplos destacam como a adoção de práticas DevOps pode oferecer benefícios significativos, incluindo entrega contínua, maior eficiência operacional, melhor colaboração entre equipes e maior confiabilidade de sistemas, em organizações de diversos setores.



FINALIZANDO

O DevOps é uma abordagem que integra as etapas de desenvolvimento e operações de *software* buscando superar desafios tradicionais por meio de automação, colaboração e entrega contínua. Além disso, envolve uma cultura centrada em princípios e valores que promovem a eficiência e a resposta ágil às demandas do mercado. A aplicação prática do DevOps é evidenciada em casos de sucesso, destacando sua evolução e relevância na vida real.

Nas próximas unidades, vamos mergulhar cada vez mais em conceitos e aplicações, considerando a base teórica abordada nesta primeira etapa.



REFERÊNCIAS

BUSINESS value of Google Cloud Dev Ops solutions. **Evonance**, [S.d.]. Disponível em: <<https://www.evonance.com/blog/business-value-of-google-cloud-dev-ops-solutions>>. Acesso em: 15 abr. 2024.

DAHIBHATE, S. DevOps Project 04. **AWS in Plain English**, 4 ago. 2023. Disponível em: <<https://aws.plainenglish.io/devops-project-04-c4e1646b267d>>. Acesso em: 15 abr. 2024.

DEFININDO a implementação distribuível em DevOps. **iMasters**, 28 jan. 2013. Disponível em: <<https://imasters.com.br/desenvolvimento/definindo-a-implementacao-distribuivel-em-devops>>. Acesso em: 15 abr. 2024.

ENAP – Escola Nacional de Administração Pública. **O papel do DevOps na transformação digital dos serviços públicos**. Brasília, 2020. Curso aberto.

KIM, G. et al. **The DevOPS Handbook**: How to Create World-Class Agility, Reliability, and Security in Technology Organizations. [S.l.: s.n.], 2016.

O QUE É DevOps?: conceitos básicos da AWS. **AWS**: Amazon Web Services, [S.d.]. Disponível em: <<https://aws.amazon.com/pt/devops/what-is-devops/>>. Acesso em: 15 abr. 2024.

PRESSMAN, R. S.; MAXIM, B. **Software engineering**: a practitioner's approach. 8. ed. [S.l.]: McGraw-Hill, 2014.

SENAC DIVINÓPOLIS. Aquecido no Brasil, mercado de TI indica aumento de vagas de trabalho. **G1**, 15 jun. 2022. Disponível em: <<https://g1.globo.com/mg/centro-oeste/especial-publicitario/senac/senac-em-divinopolis/noticia/2022/06/15/aquecido-no-brasil-mercado-de-ti-indica-aumento-de-vagas-de-trabalho.ghtml>>. Acesso em: 15 abr. 2024.